Wireless Network-Coded Three-Way Relaying Using Latin Cubes

Srishti Shukla †, Vijayvaradharaj T Muralidharan# and B. Sundar Rajan†

Email: {srishti, tmvijay, bsrajan} @ece.iisc.ernet.in
†IISc Mathematics Initiative (IMI), Dept. of Mathematics and Dept. of Electrical Comm. Engg., IISc, Bangalore
Dept. of Electrical Comm. Engg., IISc, Bangalore

***Abstract*—The design of modulation schemes for the physical layer network-coded three-way wireless relaying scenario is considered. The protocol employs two phases: Multiple Access (MA) phase and Broadcast (BC) phase with each phase utilizing one channel use. For the two-way relaying scenario, it was observed by Koike-Akino et al. [4], that adaptively changing the network coding map used at the relay according to the channel conditions greatly reduces the impact of multiple access interference which occurs at the relay during the MA phase and all these network coding maps should satisfy a requirement called *exclusive law*. This paper does the equivalent for the three-way relaying scenario. We show that when the three users transmit points from the same 4-PSK constellation, every such network coding map that satisfies the exclusive law can be represented by a Latin Cube of Second Order. The network code map used by the relay for the BC phase is explicitly obtained and is aimed at reducing the effect of interference at the MA stage.**

I. Background And Preliminaries

The concept of physical layer network coding has attracted a lot of attention in recent times. The idea of physical layer network coding for the two-way relay channel was first introduced in [1], where the multiple access interference occurring at the relay was exploited so that the communication between the end nodes can be done using a two stage protocol. Information theoretic studies for the physical layer network coding scenario were reported in [2], [3]. The design principles governing the choice of modulation schemes to be used at the nodes for uncoded transmission were studied in [4]. An extension for the case when the nodes use convolutional codes was done in [5]. A multi-level coding scheme for the two-way relaying was proposed in [6].

We consider the three-way wireless relaying scenario shown in Fig. 1, where three-way data transfer takes place among the nodes A, B and C with the help of the relay R. It is assumed that the three nodes operate in half-duplex mode. The relaying protocol consists of two phases, *multiple access* (MA) phase, consisting of one channel use during which A, B and C transmit to R; and *broadcast* (BC) phase, in which R transmits to A, B and C in a single channel use. Network Coding is employed at R in such a way that A(/B/C) can decode B's and C's(/A's and C's /A's and B's) messages, given that A(/B/C) knows its own message.

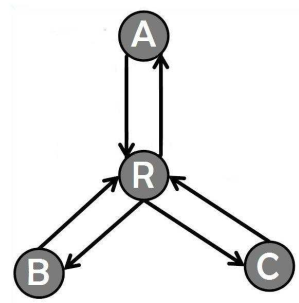

Fig. 1. A three-way relay channel

For a two-way wireless relay channel, it was observed in [4] for 4-PSK, that for uncoded transmission, the network coding map used at the relay needs to be changed adaptively according to the channel fade coefficient, in order to minimize the impact of multiple access interference. In other words, the set of all possible channel realizations is quantized into a finite number of regions, with a specific network coding map giving the best performance in a particular region. It is shown in [7] for every M-PSK constellation used by both the users, that every such network coding map that satisfies the *exclusive law* is representable as a Latin square and this relationship can be used to get the network coding maps satisfying the exclusive law. A Latin Square of order M is defined to be an $M \times M$ array in which each cell contains a symbol from $\mathbb{Z}_t = \{0, 1, ...t-1\}$ such that each symbol occurs at most once in each row and column [8].

While most of the research has been done for two-way relay channels, some work has been done for the relay channels with three or more user nodes as well. Liu and Arapostathis [9], proposed a joint network coding and superposition coding for three user relay channels, and claim that the results can be easily extended to information relaying cases with more than three relay nodes. In this scheme, two stage operations are required for encoding and decoding, and four channel uses are required for the information exchange, three channel uses for the MA phase, and one channel use for the BC phase, while for our scheme totally two channel uses suffice. In [9], the first three channel uses are utilized by each user node transmitting its packet to the relay node. Then the relay

makes two XOR-ed packets and superimposes them together for broadcast. The relay makes two XOR-ed packets, the packet from the node with the worse channel gain is XOR-ed respectively with the other two packets. Pischella and Ruyet [10] also discuss the three-way wireless relaying scenario, and propose a method of information exchange among the users composed of alternate MA and BC phases. The physical layer network coding strategy for this relaying protocol is a lattice-based coding scheme combined with power control, so that the relay receives an integer linear combination of the symbols transmitted by the user nodes. This scheme also, however, consists of four channel uses. Park and Oh, in [11], propose a network coding scheme for the three-way relay channels and present a 'Latin square-like condition' for the three-way network code. They also discuss schemes in order to improve these codes using cell swapping techniques. In this work, though Latin Cubes have been suggested as being equivalent to the map used by the relay, the number of channel uses the scheme uses is five, and the work doesn't deal with the channel gains associated with the channels explicitly. In [12], authors Jeon et al. adopt an 'opportunistic scheduling technique' for physical network coding where users in the MA as well as the BC phase are selected on the basis of instantaneous SNR using a channel norm criterion and a minimum distance criterion and plot graphs to show that the proposed scheme outperforms systems without this selection. Their approach, however, utilizes six channel uses.

For our physical layer network coding strategy, we use a mathematical structure called a Latin Cube, that has three dimensions out of which one is represented along the rows, one along the columns, and the third dimension is represented along 'files'. In our case, we have A's transmitted symbol along the files, B's symbol along the rows, and C's symbol along the columns. For our purposes, we define Latin Cubes as follows:

Definition 1: A *Latin Cube L of second order of side M* on the symbols from the set $\mathbb{Z}_t = \{0, 1, 2, ..., t-1\}$ is an $M \times M \times M$ array, in which each cell contains one symbol and each symbol occurs at most once in each row, column and file.

The above definition, is given in [13] with $t = M^2$.

A. Signal Model

Multiple Access Phase:

Let $\mathcal{S}$ denote the symmetric 4-PSK constellation $\{\pm 1, \ \pm j\}$ as shown in Fig. 2, used at A, B and C. Assume that A(/B/C) wants to send a 2-bit binary tuple to B and C(/A and C/A and B). Let $\mu : \mathbb{F}_2^2 \rightarrow \mathcal{S}$ denote the mapping from bits to complex symbols used at A, B and C where $\mathbb{F}_2 = \{0, 1\}$. Let $x_A = \mu(s_A), x_B = \mu(s_B), x_C = \mu(s_C) \in \mathcal{S}$ denote the complex symbols transmitted by A, B and C respectively, where $s_A, s_B, s_C \in \mathbb{F}_2^2$. It is assumed that the channel state information is not available at the transmitting nodes A, B and C during the MA phase. The received signal at R in the MA phase is given by

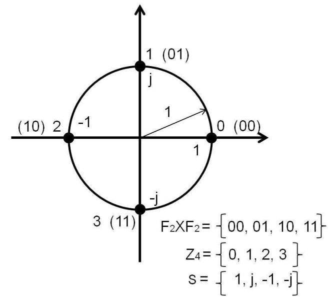

Fig. 2. 4-PSK constellation

$$Y_R = H_A x_A + H_B x_B + H_C x_C + Z_R \tag{1}$$

where H_A, H_B and H_C are the fading coefficients associated with the A-R, B-R and C-R link respectively. The additive noise Z_R is assumed to be $\mathcal{CN}(0, \sigma^2)$, where $\mathcal{CN}(0, \sigma^2)$ denotes the circularly symmetric complex Gaussian random variable with variance σ^2.

Let $\mathcal{S}_R(H_A, H_B, H_C)$ denote the effective constellations seen at the relay during the MA phase channel use, i.e.,

$$\mathcal{S}_R(H_A, H_B, H_C) = \{H_A x_i + H_B x_j + H_C x_k | x_i, x_j, x_k \in \mathcal{S}\}.$$

Let $d_{min}(H_A, H_B, H_C)$ denote the minimum distance between the points in the constellation $\mathcal{S}_R(H_A, H_B, H_C)$ as given in (2), where $\mathcal{S}^n = \mathcal{S} \times \mathcal{S} \times .. \times \mathcal{S} \ (n \, times)$. From (2), it is clear that there exists values of (H_A, H_B, H_C), for which $d_{min}(H_A, H_B, H_C) = 0$. Let $\mathcal{H} = \left\{(H_A, H_B, H_C) \in \mathbb{C}^3 | d_{min}(H_A, H_B, H_C) = 0\right\}$. The elements of $\mathcal{H}$ are called singular fade states. For singular fade states, $|\mathcal{S}_R(H_A, H_B, H_C)| < 4^3$.

Definition 2: A fade state (H_A, H_B, H_C) is defined to be a *singular fade state* for the MA phase of three-way relaying, if the cardinality of the signal set $\mathcal{S}_R(H_A, H_B, H_C)$ is less than 4^3. Let $\mathcal{H}$ denote the set of all singular fade states for the three-way data transfer among A, B and C.

Let $(\hat{x}_A, \hat{x}_B, \hat{x}_C) \in \mathcal{S}^3$ denote the Maximum Likelihood (ML) estimate of (x_A, x_B, x_C) at R based on the received complex number Y_R, i.e.,

$$(\hat{x}_A, \hat{x}_B, \hat{x}_C) = \arg \min_{(x_A, x_B, x_C) \in \mathcal{S}^3} \|Y_R - HX\| \tag{9}$$

where,

$$H = [H_A \ H_B \ H_C]$$

$$X = \begin{bmatrix} x_A \\ x_B \\ x_C \end{bmatrix}.$$

Broadcast (BC) Phase:

The received signals at A, B and C during the BC phase

$$d_{min}(H_A,H_B,H_C) = \min_{\substack{(x_A,x_B,x_C),(x'_A,x'_B,x'_C)\in\mathcal{S}^3\\(x_A,x_B,x_C)\neq(x'_A,x'_B,x'_C)}} \left|H_A\left(x_A-x'_A\right)+H_B\left(x_B-x'_B\right)+H_C\left(x_C-x'_C\right)\right| \tag{2}$$

$$d_{min}^{\mathcal{L}_i,\mathcal{L}_j}\left(H_A,H_B,H_C\right) = \min_{\substack{(x_A,x_B,x_C)\in\mathcal{L}_i,\\(x'_A,x'_B,x'_C)\in\mathcal{L}_j}} \left|H_A\left(x_A-x'_A\right)+H_B\left(x_B-x'_B\right)+H_C\left(x_C-x'_C\right)\right| \tag{3}$$

$$d_{min}\left(\mathcal{C}^{H_A,H_B,H_C}\right) = \min_{\substack{(x_A,x_B,x_C),(x'_A,x'_B,x'_C)\in\mathcal{S}^3,\\ \mathcal{M}^{H_A,H_B,H_C}(x_A,x_B,x_C)\neq\mathcal{M}^{H_A,H_B,H_C}(x'_A,x'_B,x'_C)}} \left|H_A\left(x_A-x'_A\right)+H_B\left(x_B-x'_B\right)+H_C\left(x_C-x'_C\right)\right|. \tag{4}$$

$$\mathcal{M}^{z_1,z_2}\left(x_A,x_B,x_C\right)\neq\mathcal{M}^{z_1,z_2}\left(x_A,x'_B,x'_C\right), \forall x_A,x_B,x'_B,x_C,x'_C\in\mathcal{S},\ whenever\ \left(x_B,x_C\right)\neq\left(x'_B,x'_C\right) \tag{5}$$

$$\mathcal{M}^{z_1,z_2}\left(x_A,x_B,x_C\right)\neq\mathcal{M}^{z_1,z_2}\left(x'_A,x_B,x'_C\right), \forall x_A,x'_A,x_B,x_C,x'_C\in\mathcal{S},\ whenever\ \left(x_A,x_C\right)\neq\left(x'_A,x'_C\right) \tag{6}$$

$$\mathcal{M}^{z_1,z_2}\left(x_A,x_B,x_C\right)\neq\mathcal{M}^{z_1,z_2}\left(x'_A,x'_B,x_C\right), \forall x_A,x'_A,x_B,x'_B,x_C\in\mathcal{S},\ whenever\ \left(x_A,x_B\right)\neq\left(x'_A,x'_B\right) \tag{7}$$

$$d_{min}\left(\mathcal{C}^{h_A,h_B,h_C},H_A,H_B,H_C\right) = \min_{\substack{(x_A,x_B,x_C),(x'_A,x'_B,x'_C)\in\mathcal{S}^3,\\ \mathcal{M}^{h_A,h_B,h_C}(x_A,x_B,x_C)\neq\mathcal{M}^{h_A,h_B,h_C}(x'_A,x'_B,x'_C)}} \left|H_A\left(x_A-x'_A\right)+H_B\left(x_B-x'_B\right)+H_C\left(x_C-x'_C\right)\right|. \tag{8}$$

are respectively given by,

$$Y_A = H^{'}_A X_R + Z_A,\ Y_B = H^{'}_B X_R + Z_B,\ Y_C = H^{'}_C X_R + Z_C \tag{10}$$

where $X_R = \mathcal{M}^{H_A,H_B,H_C}\left((\hat{x}_A,\hat{x}_B,\hat{x}_C)\right) \in \mathcal{S}^{'}$ is the complex number transmitted by R. The fading coefficients corresponding to the R-A, R-B and R-C links are given by $H^{'}_A$, $H^{'}_B$, and $H^{'}_C$ respectively and the additive noises Z_A, Z_B and Z_C are $\mathcal{CN}\left(0,\sigma^2\right)$. Depending on the values of H_A, H_B and H_C, R chooses a many to one map $\mathcal{M}^{H_A,H_B,H_C} : \mathcal{S}^3 \rightarrow \mathcal{S}^{'}$ where $\mathcal{S}^{'}$ is a signal set of size between 4^2 and 4^3 used by R during *BC* phase. Notice that the minimum required size for $\mathcal{S}^{'}$ is 16, since 4 bits about the other two users needs to be conveyed to each of A, B and C.

The elements in $\mathcal{S}^3$ which are mapped to the same signal point in $\mathcal{S}^{'}$ by the map $\mathcal{M}^{H_A,H_B,H_C}$ are said to form a cluster. Let $\{\mathcal{L}_1,\mathcal{L}_2,..,\mathcal{L}_l\}$ denote the set of all such clusters. The formation of clusters is called clustering, denoted by $\mathcal{C}^{H_A,H_B,H_C}$.

Definition 3: The cluster distance between a pair of clusters $\mathcal{L}_i$ and $\mathcal{L}_j$ is the minimum among all the distances calculated between the points $(x_A,x_B,x_C)\in\mathcal{L}_i$ and $(x'_A,x'_B,x'_C)\in\mathcal{L}_j$ in the effective constellation seen at the relay node R, as given in (3) above.

Definition 4: The *minimum cluster distance* of the clustering $\mathcal{C}^{H_A,H_B,H_C}$ is the minimum among all the cluster distances, as given in (4) at the top of this page.

In order to ensure that A(/B/C) is able to decode B's and C's(/A's and C's /A's and B's) message, the clustering $\mathcal{C}$ should satisfy the exclusive law, as given in (5), (6), (7) above.

The minimum cluster distance determines the performance during the MA phase of relaying. The performance during the BC phase is determined by the minimum distance of the signal set $\mathcal{S}^{'}$. For values of (H_A,H_B,H_C) in the neighborhood of the singular fade states, the value of $d_{min}\left(\mathcal{C}^{H_A,H_B,H_C}\right)$ is greatly reduced, a phenomenon referred to as *distance shortening* [4]. To avoid distance shortening, for each singular fade coefficient, a clustering needs to be chosen such that the minimum cluster distance at the singular fade state is non zero and is also maximized.

A clustering $\mathcal{C}^{H_A,H_B,H_C}$ is said to remove singular fade state $(H_A,H_B,H_C)\in\mathcal{H}$, if $d_{min}\left(\mathcal{C}^{H_A,H_B,H_C}\right)>0$. For a singular fade state $(H_A,H_B,H_C)\in\mathcal{H}$, let $\mathcal{C}^{\{(H_A,H_B,H_C)\}}$ denote the clustering which removes the singular fade state (H_A,H_B,H_C) (if there are multiple clusterings which remove the same singular fade state (H_A,H_B,H_C), consider a clustering which maximizes the minimum cluster distance). Let $\mathcal{C}_{\mathcal{H}} = \left\{\mathcal{C}^{\{(H_A,H_B,H_C)\}} : (H_A,H_B,H_C)\in\mathcal{H}\right\}$ denote the set of all such clusterings.

Definition 5: The minimum cluster distance of the clustering $\mathcal{C}^{h_A,h_B,h_C}$, when the fade state (H_A,H_B,H_C) occurs in the MA phase, denoted by $d_{min}\left(\mathcal{C}^{h_A,h_B,h_C},H_A,H_B,H_C\right)$, is the minimum among all its cluster distances, as given in (8).

For $(H_A,H_B,H_C)\notin\mathcal{H}$, the clustering $\mathcal{C}^{H_A,H_B,H_C}$ is chosen to be $\mathcal{C}^{\{(h_A,h_B,h_C)\}}$, which satisfies $d_{min}\left(\mathcal{C}^{\{(h_A,h_B,h_C)\}},H_A,H_B,H_C\right) \geq d_{min}\left(\mathcal{C}^{\left\{\left(h'_A,h'_B,h'_C\right)\right\}},H_A,H_B,H_C\right), \forall\left(h_A,h_B,h_C\right)\neq\left(h'_A,h'_B,h'_C\right)\in\mathcal{H}$. In [7], such clusterings that remove singular fade states are obtained with the help of Latin Squares while concentrating only on the first minimum cluster distance. The clustering used by the relay is indicated to A, B and C using overhead bits.

The contributions of this paper are as follows:

- Using our proposed scheme, exchange of information in the wireless three-way relaying scenario is made possible with totally two channel uses.
- It is shown that if the three users A, B, C transmit points from the same 4-PSK constellation, the requirement of satisfying the exclusive law is same as the clustering

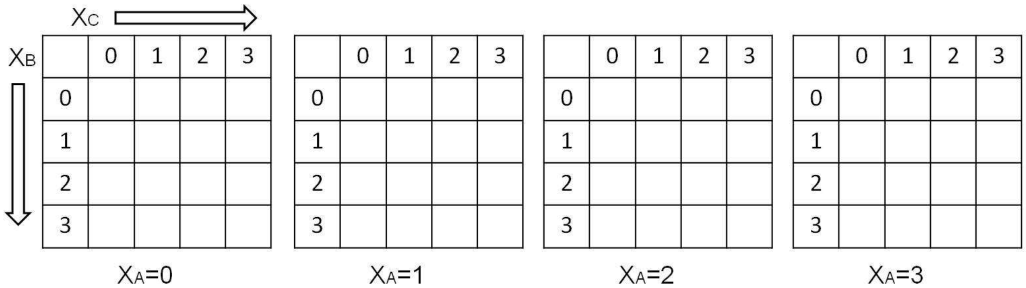

Fig. 3. The mapping observed at the Relay can be viewed as a Latin Cube of Second Order

being represented by a Latin Cube of second order of side 4. To the best of our knowledge, this is the first work with only two channel uses for the three-way relaying scenario.

- The singular fade states for the three user case are identified.
- Clusterings that removes these singular fade states are obtained, that result in the size of the constellation used by the relay node R in the BC phase to lie between 16 to 23.
- Simulation results are provided to verify that the adaptive clustering as obtained in the paper indeed performs better than non-adaptive clustering.

The remaining content is organized as follows: Section II demonstrates how a Latin Cube of Second Order and side 4 can be utilized to represent the network code for three user communication. In Section III the singular fade states are specified and in Section IV, clusterings corresponding to removal of each singular fade state are obtained using Latin Cubes of Second Order. Simulation results are shown in Section V. Section VI concludes the paper.

II. The Exclusive Law and Latin Cubes

The nodes A, B and C transmit symbols from the same constellation, viz., 4-PSK. Our aim is to find the map that relay node R should use in order to cluster the 4^3 possibilities of (x_A, x_B, x_C) such that the exclusive law (given by (5), (6), (7)) is satisfied. The size of the constellation, or the number of clusters of the clustering that the relay utilizes has to be at least 16, since each user needs the 4 bit information corresponding to the other two users. Consider a $4 \times 4 \times 4$ array, whose 64 entries are indexed by (x_A, x_B, x_C), i.e. the three messages that A, B and C send in the MA phase. Each file of this $4 \times 4 \times 4$ array, is indexed by a single value of x_A. Each row (column) of each file is indexed by a value of x_B (x_C), for a fixed value of x_A. Now, a repetition of a symbol in a file results in the failure of exclusive law given by (5). Consider the 4×4 array with its rows being the first(/second/third/forth) rows of the $4 \times 4 \times 4$ array. Each 4×4 array so obtained, corresponds to a single value of x_B. A repetition of a symbol in this array will result in the failure of exclusive law given by (6). Similarly, consider the 4×4 array with its columns being the first(/second/third/forth) columns of the $4 \times 4 \times 4$ array. Each 4×4 array so obtained, corresponds to a single value of x_C. A repetition of a symbol in this array will result in the failure of exclusive law given by (7). Hence, if the exclusive law needs to be satisfied, then the cells of this array should be filled such that the $4 \times 4 \times 4$ array so obtained, is a Latin cube of second order, for $t \geq 16$ (Definition 1). The clusters are obtained by putting together all the tuples $(i, j, k), i, j, k \in 0, 1, ...t-1$ such that the entry in the (i, j, k)-th slot is the same entry from $\mathbb{Z}_t$. From above, we can say that all the relay clusterings that satisfy the mutually exclusive law forms Latin Cubes of second order of side 4 for $t \geq 16$, when the end nodes use 4-PSK constellations. Hence, now onwards, we consider the network code used by the relay node in the BC phase to be a $4 \times 4 \times 4$ array with files(/rows/columns) being indexed by the constellation point used by A(/B/C), symbols from the set $\mathbb{Z}_4$ (Fig. 3). The cells of the array will be filled with elements of $\mathbb{Z}_t$ in such a way, that the resulting array is a Latin Cube of Second Order of side 4 and $t \geq 16$. Any arbitrary but unique symbol from $\mathbb{Z}_t$ denotes a unique cluster of a particular clustering.

III. Singular Fade Subspaces

We earlier stated in Section II, that a clustering $\mathcal{C}^{H_A, H_B, H_C}$ is said to remove singular fade state $(H_A, H_B, H_C) \in \mathcal{H}$, if $d_{min}\left(\mathcal{C}^{H_A, H_B, H_C}\right) > 0$. Alternatively, removing singular fade states for a three-way relay channel can also be defined as follows:

Definition 6: A clustering $\mathcal{C}^{H_A, H_B, H_C}$ is said to *remove the singular fade state* $(H_A, H_B, H_C) \in \mathcal{H}$, if any two possibilities of the messages sent by the users $(x_A, x_B, x_C), (x'_A, x'_B, x'_C) \in \mathcal{S}^3$ that satisfy

$$H_A x_A + H_B x_B + H_C x_C = H_A x'_A + H_B x'_B + H_C x'_C$$

are placed together in the same cluster by the clustering.

Definition 7: A set $\{(x_A, x_B, x_C)\} \in \mathcal{S}^3$ consisting of all the possibilities of (x_A, x_B, x_C) that must be placed in the same cluster of the clustering used at relay node R in the BC phase in order to remove the singular fade

state (H_A, H_B, H_C) is referred to as a *Singularity Removal Constraint* for the fade state (H_A, H_B, H_C) for three-way relaying scenario.

Let (H_A, H_B, H_C) be the fade coefficient in the MA phase. The work in [4] and [7] shows that for the two-way relaying scenario, the 4^2 possible pairs of symbols from 4-PSK constellation sent by the two users in the MA phase, can be clustered into a clustering dependent on a singular fade coefficient, of size 4 or 5 in a manner so as to remove this singular fade coefficient. In the case of three users, at the end of MA phase, relay receives a complex number, given by (1). Instead of R transmitting a point from the 4^3 point constellation resulting from all the possibilities of (x_A, x_B, x_C), the relay R can choose to group these possibilities into clusters represented by a smaller constellation. We describe one such clustering in the following.

Let Γ denote a singularity removal constraint corresponding to the singular fade state (H_A, H_B, H_C) and let $(x_A, x_B, x_C), (x'_A, x'_B, x'_C) \in \mathcal{C}$. Then,

$$\begin{aligned} &H_A x_A + H_B x_B + H_C x_C = H_A x'_A + H_B x'_B + H_C x'_C \\ \Rightarrow &H_A(x_A - x'_A) + H_B(x_B - x'_B) + H_C(x_C - x'_C) = 0 \\ \Rightarrow &(H_A, H_B, H_C) \in \left\langle \begin{bmatrix} x_A - x'_A \\ x_B - x'_B \\ x_C - x'_C \end{bmatrix} \right\rangle^{\perp} \end{aligned} \quad (11)$$

where for a 3×1 non-zero vector v over $\mathbb{C}$,

$$\langle v \rangle^{\perp} = \{ w = (w_1, w_2, w_3) \mid w_1 v_1 + w_2 v_2 + w_3 v_3 = 0 \}.$$

Clearly, $\langle v \rangle^{\perp}$ is a two-dimensional vector space of $\mathbb{C}^3$. These values of $x_A, x_B, x_C, x'_A, x'_B, x'_C \in \mathcal{S}$ result in only finitely many possibilities for the right-hand side, since $\mathcal{S}$ is finite. Thus the singular fade states (H_A, H_B, H_C), which are uncountably infinite, are points in a finite number of vector subspaces of $\mathbb{C}^3$. Henceforth, we shall refer to these finite number of vector subspaces as the *Singular Fade Subspaces*. More precisely, there are three possibilities of singular fade subspaces for the three-way relaying as we explain individually in the following three cases.

Case 1: One of the following subcases arise:

1) $x_A = x'_A,\ x_B = x'_B$ and $x_C \neq x'_C$
2) $x_A = x'_A,\ x_B \neq x'_B$ and $x_C = x'_C$
3) $x_A \neq x'_A,\ x_B = x'_B$ and $x_C = x'_C$

Case 2: One of the following subcases arise:

1) $x_A = x'_A,\ x_B \neq x'_B$ and $x_C \neq x'_C$
2) $x_A \neq x'_A,\ x_B = x'_B$ and $x_C \neq x'_C$
3) $x_A \neq x'_A,\ x_B \neq x'_B$ and $x_C = x'_C$

Case 3: $x_A \neq x'_A,\ x_B \neq x'_B$ and $x_C \neq x'_C$

Case 1: Without loss of generality, we discuss the third subcase of Case 1, i.e., the case when $x_A \neq x'_A,\ x_B = x'_B$ and $x_C = x'_C$. The singular fade subspace in this case is given by $\mathcal{S}' = \left\langle \begin{bmatrix} x_A - x'_A \\ 0 \\ 0 \end{bmatrix} \right\rangle^{\perp}$.

The set of differences of the points of $\mathcal{S}$ is given by,

$$\mathcal{D} = \{x_i - x_j \mid x_i, x_j \in \mathcal{S}\} = \{0, \pm1 \pm j, \pm2j, \pm2\}.$$

Let $\mathcal{D}_1 = \{\pm1 \pm j\}$ and $\mathcal{D}_2 = \{\pm2j, \pm2\}$. Then,

$$\mathcal{D} = \{0\} \cup \mathcal{D}_1 \cup \mathcal{D}_2.$$

Thus, $x_A - x'_A \in \mathcal{D}$ can take eight non-zero values. As a result, there are eight total possibilities for the vector $[x_A - x'_A,\ 0,\ 0]^t$, where v^t denotes the transpose of a vector v. Also, each one of $\pm1 \pm j$, $\pm2j$ and ±2 can be obtained as scalar multiples of $1 + j$ (over $\mathbb{C}$). Thus,

$$\left\langle \begin{bmatrix} 1+j \\ 0 \\ 0 \end{bmatrix} \right\rangle = \left\langle \begin{bmatrix} \pm1 \pm j \\ 0 \\ 0 \end{bmatrix} \right\rangle = \left\langle \begin{bmatrix} \pm2j \\ 0 \\ 0 \end{bmatrix} \right\rangle = \left\langle \begin{bmatrix} \pm2 \\ 0 \\ 0 \end{bmatrix} \right\rangle.$$

So, $\exists$ only one singular fade subspace for the subcase, viz.,

$$\mathcal{S}' = \left\langle \begin{bmatrix} 1+j \\ 0 \\ 0 \end{bmatrix} \right\rangle^{\perp}.$$

Similarly, for the other two subcases, there is one singular fade subspace, resulting in a total of 3 singular fade subspaces for the case.

Case 2: Without loss of generality, we discuss the third subcase of Case 2, i.e., the case when $x_A \neq x'_A,\ x_B \neq x'_B$ and $x_C = x'_C$. The singular fade subspace in this case is given by $\mathcal{S}'' = \left\langle \begin{bmatrix} x_A - x'_A \\ x_B - x'_B \\ 0 \end{bmatrix} \right\rangle^{\perp}$.

Here $x_A - x'_A$ and $x_B - x'_B \in \mathcal{D}$ can take eight non-zero values each. There are therefore, 64 total possibilities for the vector $[x_A - x'_A,\ x_B - x'_B,\ 0]^t$.

Lemma 1: For the case when $x_A - x'_A \neq 0,\ x_B - x'_B \neq 0$ and $x_C - x'_C = 0$, for a given vector $v = [x_A - x'_A,\ x_B - x'_B,\ 0]^t$ over $\mathcal{D}_1 \cup \mathcal{D}_2$, there are precisely 4 or 8 vectors (including v) over $\mathcal{D}_1 \cup \mathcal{D}_2$ that generate the same vector space over $\mathbb{C}$ as v.

Proof: As given in Section II of [7], for the 4-PSK constellation $\mathcal{S}$, the difference constellation $\mathcal{D} = \Delta\mathcal{S} = \{s - s' : s, s' \in \mathcal{S}\}$ is of the form,

$$\begin{aligned} \Delta\mathcal{S} = \{0\} &\cup \left\{ 2sin\left(\pi n/4\right) e^{jk\pi/2} | n \text{ odd} \right\} \\ &\cup \left\{ 2sin\left(\pi n/4\right) e^{j(k\pi/2 + \pi/4)} | n \text{ even} \right\}, \end{aligned}$$

where $1 \leq n \leq 2$ and $0 \leq k \leq 3$. Therefore, we can write,

$$v = \begin{bmatrix} x_A - x'_A \\ x_B - x'_B \\ 0 \end{bmatrix} = \begin{bmatrix} 2sin\frac{\pi k_1}{4}e^{j\phi_1} \\ 2sin\frac{\pi k_2}{4}e^{j\phi_2} \\ 0 \end{bmatrix}$$

where $\phi_i = k_i\pi/2$ if k_i is odd and $\phi_i = k_i\pi/2 + \pi/4$ if k_i is even.

A vector w over $\mathcal{D}_1 \cup \mathcal{D}_2$ shall generate the same vector space over $\mathbb{C}$ iff w is a scalar multiple of v, i.e. for some complex number $re^{j\theta} \in \mathbb{C}$,

$$v = re^{j\theta}w \Rightarrow \begin{bmatrix} 2sin\frac{\pi k_1}{4}e^{j\phi_1} \\ 2sin\frac{\pi k_2}{4}e^{j\phi_2} \\ 0 \end{bmatrix} = re^{j\theta} \begin{bmatrix} 2sin\frac{\pi k_3}{4}e^{j\phi_3} \\ 2sin\frac{\pi k_4}{4}e^{j\phi_4} \\ 0 \end{bmatrix}$$

where for $i = 3, 4$ $\phi_i = k_i\pi/2$ if k_i is odd and $\phi_i = k_i\pi/2 + \pi/4$ if k_i is even.

Then,

$$2sin\frac{\pi k_1}{4}e^{j\phi_1} = re^{j\theta} \times 2sin\frac{\pi k_3}{4}e^{j\phi_3} \tag{12}$$

and

$$2sin\frac{\pi k_2}{4}e^{j\phi_2} = re^{j\theta} \times 2sin\frac{\pi k_4}{4}e^{j\phi_4}. \tag{13}$$

Dividing (12) by (13) and taking modulus of both sides, we get

$$\frac{sin\frac{\pi k_1}{4}}{sin\frac{\pi k_2}{4}} = \frac{sin\frac{\pi k_3}{4}}{sin\frac{\pi k_4}{4}} \tag{14}$$

As shown in [7], this is possible only if $k_1 = k_3$ and $k_2 = k_4$. Also, from (12) and (13) we have

$$\frac{sin\frac{\pi k_1}{4}}{sin\frac{\pi k_2}{4}}e^{j(\phi_1-\phi_2)} = \frac{sin\frac{\pi k_3}{4}}{sin\frac{\pi k_4}{4}}e^{j(\phi_3-\phi_4)} \tag{15}$$

From (14) and (16), we have

$$e^{j(\phi_1-\phi_2)} = e^{j(\phi_3-\phi_4)} \tag{16}$$

Note that here, the LHS is fixed. It therefore suffices to compute that for the fixed value of the LHS, the number of values that RHS takes. It can be verified, that for a fixed value of $\phi_1 - \phi_2$, there are precisely four pair of values of ϕ_3 and ϕ_4 that result in the same value of $\phi_3 - \phi_4$. We now look at the following possibilities:

Case 1: $k_1 = k_2$. Then, $k_3 = k_4$, i.e., there are exactly two possibilities for k_1 and k_2, viz., $k_1 = k_2 = 1$ and $k_1 = k_2 = 2$. With two pairs of values for k_3 and k_4 and four pairs of values for ϕ_3 and ϕ_4, we have a total of eight set of values that w can take. Hence, in this case, the vector space generated by v can be generated by exactly eight other vectors over $\mathcal{D}_1 \cup \mathcal{D}_2$.

Case 2: $k_1 \neq k_2$. Then, $k_3 \neq k_4$, i.e., there is precisely one possibility for k_1 and k_2, viz., $k_1 = k_3$ and $k_1 = k_4$. With only one possible set of values for k_3 and k_4 and four pairs of values for ϕ_3 and ϕ_4, we have a total of four set of values that w can take. Hence, in this case, the vector space generated by v can be generated by exactly four other vectors over $\mathcal{D}_1 \cup \mathcal{D}_2$. ■

In this case we end up with 12 singular fade subspaces given by the null spaces of the space given on the next page in Figure III.

So, $\exists$ 12 singular fade subspaces for the subcase each being the null space of the above 12 spaces. Similarly, for each of the other two subcases, there are 12 singular fade subspaces, resulting in a total of 36 singular fade subspaces for the case.

Case 3: In this case, $x_A \neq x'_A,\ x_B \neq x'_B$ and $x_C \neq x'_C$. The singular fade subspace in this case is given by

$$\mathcal{S}''' = \left\langle \begin{bmatrix} x_A - x'_A \\ x_B - x'_B \\ x_C - x'_C \end{bmatrix} \right\rangle^{\perp}. \tag{17}$$

Here each of $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}$ can take eight non-zero values. There are therefore, 512 total possibilities for the vector $[x_A - x'_A,\ x_B - x'_B,\ x_C - x'_C]^t$. In this case we end up with 112 singular fade subspaces. We now explain this.

Lemma 2: For the case when $x_A - x'_A \neq 0,\ x_B - x'_B \neq 0$ and $x_C - x'_C \neq 0$, for a given vector $v = [x_A - x'_A,\ x_B - x'_B,\ x_C - x'_C]^t$ over $\mathcal{D}_1 \cup \mathcal{D}_2$, there are precisely 4 or 8 vectors (including v) over $\mathcal{D}_1 \cup \mathcal{D}_2$ that generate the same vector space over $\mathbb{C}$ as v.

Proof: As mentioned in the proof of Lemma 1, from [7],

$$\mathcal{D} = \Delta\mathcal{S} = \{0\} \cup \left\{2sin\left(\pi n/4\right)e^{jk\pi/2} | n \text{ odd}\right\} \\ \cup \left\{2sin\left(\pi n/4\right)e^{j(k\pi/2+\pi/4)} | n \text{ even}\right\},$$

where $1 \leq n \leq 2$ and $0 \leq k \leq 3$. Therefore, we can write,

$$v = \begin{bmatrix} x_A - x'_A \\ x_B - x'_B \\ x_C - x'_C \end{bmatrix} = \begin{bmatrix} 2sin\frac{\pi k_1}{4}e^{j\phi_1} \\ 2sin\frac{\pi k_2}{4}e^{j\phi_2} \\ 2sin\frac{\pi k_3}{4}e^{j\phi_3} \end{bmatrix}$$

where $\phi_i = k_i\pi/2$ if k_i is odd and $\phi_i = k_i\pi/2 + \pi/4$ if k_i is even.

A vector w over $\mathcal{D}_1 \cup \mathcal{D}_2$ shall generate the same vector space over $\mathbb{C}$ iff w is a scalar multiple of v, i.e. for some complex number $re^{j\theta} \in \mathbb{C}$,

$$v = re^{j\theta}w \Rightarrow \begin{bmatrix} 2sin\frac{\pi k_1}{4}e^{j\phi_1} \\ 2sin\frac{\pi k_2}{4}e^{j\phi_2} \\ 2sin\frac{\pi k_3}{4}e^{j\phi_3} \end{bmatrix} = re^{j\theta} \begin{bmatrix} 2sin\frac{\pi k_4}{4}e^{j\phi_4} \\ 2sin\frac{\pi k_5}{4}e^{j\phi_5} \\ 2sin\frac{\pi k_6}{4}e^{j\phi_6} \end{bmatrix}$$

where for $i = 4, 5, 6$ $\phi_i = k_i\pi/2$ if k_i is odd and $\phi_i = k_i\pi/2 + \pi/4$ if k_i is even.

$$1.\ \left\langle\begin{bmatrix}1+j\\1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2\\0\end{bmatrix}\right\rangle$$

$$2.\ \left\langle\begin{bmatrix}1+j\\-1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2\\0\end{bmatrix}\right\rangle$$

$$3.\ \left\langle\begin{bmatrix}1+j\\1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2j\\0\end{bmatrix}\right\rangle$$

$$4.\ \left\langle\begin{bmatrix}1+j\\-1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2j\\0\end{bmatrix}\right\rangle$$

$$5.\ \left\langle\begin{bmatrix}1+j\\2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2\\0\end{bmatrix}\right\rangle \quad 6.\ \left\langle\begin{bmatrix}1+j\\-2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2\\0\end{bmatrix}\right\rangle$$

$$7.\ \left\langle\begin{bmatrix}1+j\\2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2j\\0\end{bmatrix}\right\rangle \quad 8.\ \left\langle\begin{bmatrix}1+j\\-2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2j\\0\end{bmatrix}\right\rangle$$

$$9.\ \left\langle\begin{bmatrix}2j\\1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\1-j\\0\end{bmatrix}\right\rangle \quad 10.\ \left\langle\begin{bmatrix}-2j\\1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\1-j\\0\end{bmatrix}\right\rangle$$

$$11.\ \left\langle\begin{bmatrix}2\\1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\1-j\\0\end{bmatrix}\right\rangle \quad 12.\ \left\langle\begin{bmatrix}-2\\1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-1-j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-1+j\\0\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\1-j\\0\end{bmatrix}\right\rangle.$$

Fig. 4. Null Spaces of the Singular Fades Spaces for the case $x_A \neq x'_A,\ x_B \neq x'_B$ and $x_C = x'_C$.

Then,

$$2sin\frac{\pi k_1}{4}e^{j\phi_1} = re^{j\theta} \times 2sin\frac{\pi k_4}{4}e^{j\phi_4}, \tag{18}$$

$$2sin\frac{\pi k_2}{4}e^{j\phi_2} = re^{j\theta} \times 2sin\frac{\pi k_5}{4}e^{j\phi_5} \tag{19}$$

and,

$$2sin\frac{\pi k_3}{4}e^{j\phi_3} = re^{j\theta} \times 2sin\frac{\pi k_6}{4}e^{j\phi_6} \tag{20}$$

Dividing (18) by (19) and taking modulus of both sides, we get

$$\frac{sin\frac{\pi k_1}{4}}{sin\frac{\pi k_2}{4}} = \frac{sin\frac{\pi k_4}{4}}{sin\frac{\pi k_5}{4}} \tag{21}$$

As shown in [7], this is possible only if $k_1 = k_4$ and $k_2 = k_5$. Similarly, dividing (18) by (20) and taking modulus of both sides, we get

$$\frac{sin\frac{\pi k_1}{4}}{sin\frac{\pi k_3}{4}} = \frac{sin\frac{\pi k_4}{4}}{sin\frac{\pi k_6}{4}} \tag{22}$$

As shown in [7], this is possible only if $k_1 = k_4$ and $k_3 = k_6$. Also, from (18) and (19) we have

$$\frac{sin\frac{\pi k_1}{4}}{sin\frac{\pi k_2}{4}}e^{j(\phi_1-\phi_2)} = \frac{sin\frac{\pi k_4}{4}}{sin\frac{\pi k_5}{4}}e^{j(\phi_4-\phi_5)} \tag{23}$$

From (21) and (23), we have

$$e^{j(\phi_1-\phi_2)} = e^{j(\phi_4-\phi_5)} \tag{24}$$

Similarly,

$$e^{j(\phi_1-\phi_3)} = e^{j(\phi_4-\phi_6)} \tag{25}$$

In (24) and (25), the LHS is fixed. It therefore suffices to compute the number of values that RHS takes for fixed LHS in the two equations. It can be verified, that for a fixed value of $\phi_1,\ \phi_2$ and ϕ_3, there are precisely four set of values of $\phi_4,\ \phi_5$ and ϕ_6 that result in the same value of $\phi_1 - \phi_2$ and $\phi_1 - \phi_3$. We now look at the following possibilities:

Case 1: $k_1 = k_2 = k_3$. Then, $k_4 = k_5 = k_6$, i.e., there are exactly two possibilities for k_4, k_5 and k_6, viz., $k_4 = k_5 = k_6 = 1$ and $k_4 = k_5 = k_6 = 2$. With two sets of values for k_4, k_5 and k_6 and four sets of values for ϕ_4, ϕ_5 and ϕ_6, we have a total of eight set of values that w can take. Hence, in this case, the vector space generated by v can be generated by exactly eight other vectors over $\mathcal{D}_1 \cup \mathcal{D}_2$. *Case 2:* Atleast one of $k_1 \neq k_2$, $k_1 \neq k_3$, or $k_2 \neq k_3$. Then, $k_4 \neq k_5$, $k_4 \neq k_6$, or $k_5 \neq k_6$, so that, using (21) and (22), there is precisely one possibility for $k_4,\ k_5,\ k_6$, viz., $k_4 = k_1$, $k_5 = k_2$ and $k_6 = k_3$. With only one possible set of values for $k_4,\ k_5$ and k_6 and four sets of values for $\phi_4,\ \phi_5$ and ϕ_6, we have a total of four set of values that w can take. Hence, in this case, the vector space generated by v can be generated by exactly four other vectors over $\mathcal{D}_1 \cup \mathcal{D}_2$.

■

Since all of $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}$ are non-zero, we can say that

$$x_A - x'_A,\ x_B - x'_B \text{ and } x_C - x'_C \in \mathcal{D}_1 \cup \mathcal{D}_2.$$

As a result, we have the following three subcases:

1) One of $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}_1$
2) Two of $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}_1$
3) All of $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}_1$

We deal with each one of the subcases one-by-one.

Subcase 1: One of $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}_1$. Without loss of generality, we assume that $x_A - x'_A \in \mathcal{D}_1$ and $x_B - x'_B,\ x_C - x'_C \in \mathcal{D}_2$. The singular fade subspace for the case is given by (17). There are 64 possibilities for the vector $v' = [x_A - x'_A,\ x_B - x'_B,\ x_C - x'_C]^t$. But some of the possibilities may generate the same vector space over $\mathbb{C}$. There are precisely 4 vectors of length 3 over $\mathcal{D}_1 \cup \mathcal{D}_2$, the $\{\pm 1, \pm j\}$ scalar multiples of the vector. As a result,

the case $x_A - x'_A \in \mathcal{D}_1$ and $x_B - x'_B,\ x_C - x'_C \in \mathcal{D}_2$ leads to 16 singular fade subspaces as shown in Figure III. The same holds for the case when $x_B - x'_B \in \mathcal{D}_1$ and $x_A - x'_A,\ x_C - x'_C \in \mathcal{D}_2$, or when $x_C - x'_C \in \mathcal{D}_1$ and $x_A - x'_A,\ x_B - x'_B \in \mathcal{D}_2$. This subcase therefore results in 48 singular fade subspaces.

Subcase 2: Two of $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}_1$. Without loss of generality, we assume that $x_A - x'_A, x_B - x'_B \in \mathcal{D}_1$ and $x_C - x'_C \in \mathcal{D}_2$. The singular fade subspace for the case is given by (17) and again there are 64 possibilities for the vector $v'' = [x_A - x'_A,\ x_B - x'_B,\ x_C - x'_C]^t$. For a given v'', possibilities of other 3 length vectors over $\mathcal{D}_1 \cup \mathcal{D}_2$ that generate the same vector space over $\mathbb{C}$ are the $\{\pm 1, \pm j\}$ scalar multiples of v''. As a result, the case $x_A - x'_A \in \mathcal{D}_1$ and $x_B - x'_B,\ x_C - x'_C \in \mathcal{D}_2$ also leads to 16 singular fade subspaces as shown for this case in Figure (III). The same holds for the case when $x_B - x'_B \in \mathcal{D}_2$ and $x_A - x'_A,\ x_C - x'_C \in \mathcal{D}_1$, or when $x_A - x'_A \in \mathcal{D}_2$ and $x_B - x'_B,\ x_C - x'_C \in \mathcal{D}_1$ resulting in therefore 48 singular fade subspaces.

Subcase 3: All of $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}_1$. There are 64 possibilities for the vector $[x_A - x'_A,\ x_B - x'_B,\ x_C - x'_C]^t$ over $\mathcal{D}_1$. For a given such vector, possibilities of other 3 length vectors over $\mathcal{D}_1 \cup \mathcal{D}_2$ that generate the same vector space over $\mathbb{C}$ are the $\{\pm 1, \pm j, \pm 1 \pm j\}$ scalar multiples of the vector. The case $x_A - x'_A,\ x_B - x'_B,\ x_C - x'_C \in \mathcal{D}_1$ leads to a total of 16 singular fade subspaces as shown in Figure (III).

The three cases result in a total of 3+36+48+48+16=151 singular fade subspaces. We now discuss how these singular fade subspaces can be removed using Latin Cubes of Second order.

IV. Removing singular fade subspaces and Constrained Latin Cubes

In the previous section, we classify the set of singular fade subspaces into three cases. We now cluster the possibilities of (x_A, x_B, x_C) into a clustering using Latin Cubes. This clustering is represented by a constellation given by $\mathcal{S}'$, which is utilized by the relay node R in the BC phase. The objective is to minimize the size of this constellation used by R.

The clustering to be used at R, first constrains the possibilities of (x_A, x_B, x_C) received at the MA phase, with the objective of removing the singular fade subspaces, and fills the entries of a $4 \times 4 \times 4$ array representing the map to be used at the relay using these constraints, and then completes the partially empty array obtained to form a Latin cube of second order. The mapping to be used at R can be obtained from the complete Latin cube keeping in mind the equivalence between the relay map with the Latin Cube of second order as shown in Section III.

In order to to obtain the constraints on the $4 \times 4 \times 4$ array representing the map at the relay node R during BC phase for a singular fade state, we utilize the vectors of differences, viz., $[x_A - x'_A\ x_B - x'_B,\ x_C - x'_C]$ contributing to that particular singular fade state. During MA phase for the three-way relaying scenario, nodes A, B and C transmit to the relay R. As shown in the previous section, there is a total of 151 singular fade subspaces. Let (h_A, h_B, h_C) denote a point in one of the 151 singular fade subspaces. Then, there exist $(x_A, x_B, x_C), (x'_A, x'_B, x'_C) \in \mathcal{S}^3$ that satisfy $h_A x_A + h_B x_B + h_C x_C = h_A x'_A + h_B x'_B + h_C x'_C$. In order to remove the singular fade state (h_A, h_B, h_C), the pair $(x_A, x_B, x_C), (x'_A, x'_B, x'_C)$ must be kept in the same cluster in the clustering. For instance, if

$$(h_A, h_B, h_C) \in \left\langle \begin{bmatrix} x_A - x'_A \\ x_B - x'_B \\ x_C - x'_C \end{bmatrix} \right\rangle^{\perp},$$

then the pair $(x_A, x_B, x_C), (x'_A, x'_B, x'_C)$ must be kept in the same cluster in the clustering, i.e., the entry corresponding to (x_A, x_B, x_C) in the $4 \times 4 \times 4$ array must be the same as the entry corresponding to (x'_A, x'_B, x'_C). Similarly, every other such pair in $\mathcal{S}^3$ contributing to the same singular fade subspace must be kept in the same cluster. Apart from all such pairs in $\mathcal{S}^3$ being kept in the same cluster of the clustering, in order to remove this particular fade state, there are no other constraints. Upon filling up the $4 \times 4 \times 4$ array with these constraints, the entire Latin Cube does not get filled up, but can be completed as we show later in this section. It is important to note that, this clustering cannot be utilized to remove the singular fade subspaces of Case 1 and Case 2 of the previous section, as shown in the following lemma.

Lemma 3: The clustering map used at the relay node R cannot remove a singular fade state corresponding to the Case 1 and Case 2 of the previous section and simultaneously satisfy the mutually exclusive law.

Proof: Let $\mathcal{S}' = \left\langle \begin{bmatrix} x_A - x'_A \\ x_B - x'_B \\ x_C - x'_C \end{bmatrix} \right\rangle^{\perp}$ be a singular fade state for Case 1, and without loss of generality, assume that $x_B = x'_B$ and $x_C = x'_C$. Then, in order to remove $\mathcal{S}'$, (x_A, x_B, x_C) and (x'_A, x_B, x_C) must be kept in the same cluster. This is impossible, since this clearly violates the mutually exclusive law as if the pair is placed in the same cluster, the users B and C will not be able to distinguish between the messages x_A and x'_A sent by the user A.

Let $\mathcal{S}'$ be a singular fade state for Case 2, and without loss of generality, assume that $x_C = x'_C$. Then, in order to remove $\mathcal{S}'$, (x_A, x_B, x_C) and (x'_A, x'_B, x_C) must be kept in the same cluster. This also clearly violates the mutually exclusive law since if the pair is placed in the same cluster, the user C will not be able to distinguish between the messages $x_A,\ x_B$ and

$$1.\ \left\langle\begin{bmatrix}1+j\\2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2\\2\end{bmatrix}\right\rangle \quad 2.\ \left\langle\begin{bmatrix}1+j\\2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2\\-2\end{bmatrix}\right\rangle$$

$$3.\ \left\langle\begin{bmatrix}1+j\\2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2\\-2j\end{bmatrix}\right\rangle \quad 4.\ \left\langle\begin{bmatrix}1+j\\2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2\\2j\end{bmatrix}\right\rangle$$

$$5.\ \left\langle\begin{bmatrix}1+j\\-2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2\\2\end{bmatrix}\right\rangle \quad 6.\ \left\langle\begin{bmatrix}1+j\\-2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2\\-2\end{bmatrix}\right\rangle$$

$$7.\ \left\langle\begin{bmatrix}21+j\\-2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2\\-2j\end{bmatrix}\right\rangle \quad 8.\ \left\langle\begin{bmatrix}1+j\\-2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2\\2j\end{bmatrix}\right\rangle.$$

$$9.\ \left\langle\begin{bmatrix}1+j\\2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2j\\2\end{bmatrix}\right\rangle \quad 10.\ \left\langle\begin{bmatrix}1+j\\2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2j\\-2\end{bmatrix}\right\rangle$$

$$11.\ \left\langle\begin{bmatrix}1+j\\2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2j\\-2j\end{bmatrix}\right\rangle \quad 12.\ \left\langle\begin{bmatrix}1+j\\2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-2j\\2j\end{bmatrix}\right\rangle$$

$$13.\ \left\langle\begin{bmatrix}1+j\\-2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2j\\2\end{bmatrix}\right\rangle \quad 14.\ \left\langle\begin{bmatrix}1+j\\-2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2j\\2\end{bmatrix}\right\rangle$$

$$15.\ \left\langle\begin{bmatrix}1+j\\-2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2j\\-2j\end{bmatrix}\right\rangle \quad 16.\ \left\langle\begin{bmatrix}1+j\\-2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\2j\\2j\end{bmatrix}\right\rangle.$$

Fig. 5. Null Spaces of the Singular Fades Spaces for the case $x_A - x'_A \in \mathcal{D}_1$ and $x_B - x'_B,\ x_C - x'_C \in \mathcal{D}_2$.

$$1.\ \left\langle\begin{bmatrix}1+j\\1+j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\2\end{bmatrix}\right\rangle \quad 2.\ \left\langle\begin{bmatrix}1+j\\1+j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\-2\end{bmatrix}\right\rangle$$

$$3.\ \left\langle\begin{bmatrix}1+j\\1+j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\-2j\end{bmatrix}\right\rangle \quad 4.\ \left\langle\begin{bmatrix}1+j\\1+j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\2j\end{bmatrix}\right\rangle$$

$$5.\ \left\langle\begin{bmatrix}1+j\\-1-j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\2\end{bmatrix}\right\rangle \quad 6.\ \left\langle\begin{bmatrix}1+j\\-1-j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\-2\end{bmatrix}\right\rangle$$

$$7.\ \left\langle\begin{bmatrix}1+j\\-1-j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\-2j\end{bmatrix}\right\rangle \quad 8.\ \left\langle\begin{bmatrix}1+j\\-1-j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\2j\end{bmatrix}\right\rangle.$$

$$9.\ \left\langle\begin{bmatrix}1+j\\1-j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\2\end{bmatrix}\right\rangle \quad 10.\ \left\langle\begin{bmatrix}1+j\\1-j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\-2\end{bmatrix}\right\rangle$$

$$11.\ \left\langle\begin{bmatrix}1+j\\1-j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\-2j\end{bmatrix}\right\rangle \quad 12.\ \left\langle\begin{bmatrix}1+j\\1-j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\2j\end{bmatrix}\right\rangle$$

$$13.\ \left\langle\begin{bmatrix}1+j\\-1+j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\2\end{bmatrix}\right\rangle \quad 14.\ \left\langle\begin{bmatrix}1+j\\-1+j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\-2\end{bmatrix}\right\rangle$$

$$15.\ \left\langle\begin{bmatrix}1+j\\-1+j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\-2j\end{bmatrix}\right\rangle \quad 16.\ \left\langle\begin{bmatrix}1+j\\-1+j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\2j\end{bmatrix}\right\rangle.$$

Fig. 6. Null Spaces of the Singular Fades Spaces for the case $x_A - x'_A,\ x_B - x'_B \in \mathcal{D}_1$ and $x_C - x'_C \in \mathcal{D}_2$.

messages $x'_A,\ x'_B$ sent by the users A and B. ■

The singular fade subspaces given in Case 1 ans Case 2 of the previous section, whose harmful effects cannot be removed by a proper choice of the clustering are referred to as the *non-removable singular fade subspaces*. The rest of the singular fade subspaces, given in Case 3 of the previous section, are referred as the *removable singular fade subspaces*.

We now illustrate the removal of a Case 3 singular fade state with the help of examples.

Example 1: Let the singular fade subspace be

$$\mathcal{S}'=\left\langle\begin{bmatrix}1+j\\2j\\-2j\end{bmatrix}\right\rangle^{\perp}=\left\langle\begin{bmatrix}-1-j\\-2j\\2j\end{bmatrix}\right\rangle^{\perp}=\left\langle\begin{bmatrix}-1+j\\-2\\2\end{bmatrix}\right\rangle^{\perp}=\left\langle\begin{bmatrix}1-j\\2\\-2\end{bmatrix}\right\rangle^{\perp}$$

Consider the first vector $[1+j,\ 2j,\ -2j]^t$. Here, $1+j$ can be obtained as a difference of $x_A=1$ and $x'_A=-j$ or as a difference of $x_A=j$ and $x'_A=-1$; $2j$ can be obtained as a difference of $x_B=j$ and $x'_B=-j$; and $-2j$ can be obtained as a difference of $x_C=-j$ and $x'_C=j$. Thus, the entries corresponding to $(1,j,-j)$ and $(-j,-j,j)$ must be the same and the entries corresponding to $(j,j,-j)$ and $(-1,-j,j)$ must be the same in the $4\times4\times4$ array representing the clustering, i.e., entries $(0,1,3)$ and $(3,3,1)$ must be the same and entries $(1,1,3)$ and $(2,3,1)$ must be the same.

The second vector $[-1-j,\ -2j,\ 2j]^t$ where, $-1-j$ can be obtained as a difference of $x_A=-1$ and $x'_A=j$ or as a difference of $x_A=-j$ and $x'_A=1$; $-2j$ can be obtained as a difference of $x_B=-j$ and $x'_B=j$; and $2j$ can be obtained as a difference of $x_C=j$ and $x'_C=-j$. Thus, the entries

$$1.\ \left\langle\begin{bmatrix}1+j\\1+j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2\\-2\end{bmatrix}\right\rangle$$

$$2.\ \left\langle\begin{bmatrix}1+j\\1+j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2\\2\end{bmatrix}\right\rangle$$

$$3.\ \left\langle\begin{bmatrix}1+j\\1+j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2\\2j\end{bmatrix}\right\rangle$$

$$4.\ \left\langle\begin{bmatrix}1+j\\1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1-j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1+j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1-j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2\\-2j\end{bmatrix}\right\rangle$$

$$5.\ \left\langle\begin{bmatrix}1+j\\-1-j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2\\-2\end{bmatrix}\right\rangle$$

$$6.\ \left\langle\begin{bmatrix}1+j\\-1-j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2\\2\end{bmatrix}\right\rangle$$

$$7.\ \left\langle\begin{bmatrix}1+j\\-1-j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2\\-2j\end{bmatrix}\right\rangle$$

$$8.\ \left\langle\begin{bmatrix}1+j\\-1-j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1-j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1+j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2\\2j\end{bmatrix}\right\rangle$$

$$9.\ \left\langle\begin{bmatrix}1+j\\1-j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2\\2j\end{bmatrix}\right\rangle$$

$$10.\ \left\langle\begin{bmatrix}1+j\\1-j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2j\\2\end{bmatrix}\right\rangle$$

$$11.\ \left\langle\begin{bmatrix}1+j\\1-j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2j\\2j\end{bmatrix}\right\rangle$$

$$12.\ \left\langle\begin{bmatrix}1+j\\1-j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\-1+j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\1+j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\-1-j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\-2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\-2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\2j\\-2j\end{bmatrix}\right\rangle$$

$$13.\ \left\langle\begin{bmatrix}1+j\\-1+j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2j\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2j\\-2\end{bmatrix}\right\rangle$$

$$14.\ \left\langle\begin{bmatrix}1+j\\-1+j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2j\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2j\\2\end{bmatrix}\right\rangle$$

$$15.\ \left\langle\begin{bmatrix}1+j\\-1+j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2j\\-2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2j\\2j\end{bmatrix}\right\rangle$$

$$16.\ \left\langle\begin{bmatrix}1+j\\-1+j\\-1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1-j\\1-j\\1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-1+j\\-1-j\\-1-j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}1-j\\1+j\\1+j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2j\\-2\\-2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2j\\2\\2\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}2\\2j\\2j\end{bmatrix}\right\rangle=\left\langle\begin{bmatrix}-2\\-2j\\-2j\end{bmatrix}\right\rangle.$$

Fig. 7. Null Spaces of the Singular Fades Spaces for the case $x_A - x'_A,\ x_B - x'_B$ and $x_C - x'_C \in \mathcal{D}_1$.

corresponding to $(-1,-j,j)$ and $(j,j,-j)$ must be the same and the entries corresponding to $(-j,-j,j)$ and $(1,j,-j)$ must be the same in the $4 \times 4 \times 4$ array representing the clustering, i.e., entries $(2,3,1)$ and $(1,1,3)$ must be the same and entries $(3,3,1)$ and $(0,1,3)$ must be the same. Similarly, the constraints resulting from the third and forth vector ca be obtained. The constrained array is shown in Figure 8.

We choose to fill the constrained array to form a Latin cube of second order as shown in adjoining Algorithm 1, which simply fills the empty cells of the partially filled $4 \times 4 \times 4$ array with $\mathcal{L}_i,\ i \geq 1$ in increasing order of i keeping in mind that the resulting array must be a Latin cube of second order. The top-most and the left-most empty cell in the earliest file is filled at every iteration. The completed Latin cube is shown in Figure 9.

Example 2: Consider another example for which the singular fade subspace is given by

$$\mathcal{S}'' = \left\langle\begin{bmatrix}1+j\\1+j\\-1-j\end{bmatrix}\right\rangle^{\perp} = \left\langle\begin{bmatrix}-1-j\\-1-j\\1+j\end{bmatrix}\right\rangle^{\perp} = \left\langle\begin{bmatrix}-1+j\\-1+j\\1-j\end{bmatrix}\right\rangle^{\perp} = \left\langle\begin{bmatrix}1-j\\1-j\\-1+j\end{bmatrix}\right\rangle^{\perp}$$
$$= \left\langle\begin{bmatrix}2j\\2j\\-2j\end{bmatrix}\right\rangle^{\perp} = \left\langle\begin{bmatrix}-2j\\-2j\\2j\end{bmatrix}\right\rangle^{\perp} = \left\langle\begin{bmatrix}-2\\-2\\2\end{bmatrix}\right\rangle^{\perp} = \left\langle\begin{bmatrix}2\\2\\-2\end{bmatrix}\right\rangle^{\perp}.$$

The first vector is $[1+j,\ 1+j,\ -1-j]$. Here, $1+j$ can be obtained as a difference of $x_A = 1$ and $x'_A = -j$ or as a difference of $x_A = j$ and $x'_A = -1$; $-1-j$ can be obtained as a difference of $x_C = -1$ and $x'_C = j$ or as a difference of $x_C = -j$ and $x'_C = 1$. Thus, the entries corresponding to $\{(1,1,-1),(-j,-j,j)\}$, $\{(1,1,-j),(-j,-j,1)\}$,$\{(1,j,-1),(-j,-1,j)\}$,$\{(1,j,-j),(-j,-1,1)\}$, $\{(j,1,-1),(-1,-j,j)\}$,$\{(j,1,-j),(-1,-j,1)\}$,$\{(j,j,-1),(-1,-1,j)\}$, $\{(j,j,-j),(-1,-1,1)\}$ must be the same in the $4 \times 4 \times 4$ array representing the clustering, i.e., entries $\{(0,0,2),(3,3,1)\}$,

$x_A = 0$	0	1	2	3
0			$\mathcal{L}_3$	
1				$\mathcal{L}_2$
2				
3				

$x_A = 1$	0	1	2	3
0				
1				$\mathcal{L}_1$
2	$\mathcal{L}_3$			
3				

$x_A = 2$	0	1	2	3
0				
1				
2	$\mathcal{L}_4$			
3		$\mathcal{L}_1$		

$x_A = 3$	0	1	2	3
0			$\mathcal{L}_4$	
1				
2				
3		$\mathcal{L}_2$		

Fig. 8. Constraints for Example 1, with B's transmitted symbols along the rows and C's transmitted symbols along the columns

$x_A = 0$	0	1	2	3
0	$\mathcal{L}_1$	$\mathcal{L}_5$	$\boldsymbol{\mathcal{L}_3}$	$\mathcal{L}_6$
1	$\mathcal{L}_7$	$\mathcal{L}_4$	$\mathcal{L}_8$	$\boldsymbol{\mathcal{L}_2}$
2	$\mathcal{L}_9$	$\mathcal{L}_{10}$	$\mathcal{L}_{11}$	$\mathcal{L}_{12}$
3	$\mathcal{L}_{13}$	$\mathcal{L}_{14}$	$\mathcal{L}_{15}$	$\mathcal{L}_{16}$

$x_A = 1$	0	1	2	3
0	$\mathcal{L}_2$	$\mathcal{L}_7$	$\mathcal{L}_9$	$\mathcal{L}_8$
1	$\mathcal{L}_5$	$\mathcal{L}_6$	$\mathcal{L}_{10}$	$\boldsymbol{\mathcal{L}_1}$
2	$\boldsymbol{\mathcal{L}_3}$	$\mathcal{L}_{13}$	$\mathcal{L}_{14}$	$\mathcal{L}_{15}$
3	$\mathcal{L}_{11}$	$\mathcal{L}_{12}$	$\mathcal{L}_{17}$	$\mathcal{L}_4$

$x_A = 2$	0	1	2	3
0	$\mathcal{L}_{10}$	$\mathcal{L}_{11}$	$\mathcal{L}_{12}$	$\mathcal{L}_{13}$
1	$\mathcal{L}_{14}$	$\mathcal{L}_3$	$\mathcal{L}_{16}$	$\mathcal{L}_9$
2	$\boldsymbol{\mathcal{L}_4}$	$\mathcal{L}_8$	$\mathcal{L}_2$	$\mathcal{L}_5$
3	$\mathcal{L}_6$	$\boldsymbol{\mathcal{L}_1}$	$\mathcal{L}_7$	$\mathcal{L}_{18}$

$x_A = 3$	0	1	2	3
0	$\mathcal{L}_{15}$	$\mathcal{L}_{16}$	$\boldsymbol{\mathcal{L}_4}$	$\mathcal{L}_{14}$
1	$\mathcal{L}_{12}$	$\mathcal{L}_{17}$	$\mathcal{L}_{13}$	$\mathcal{L}_{11}$
2	$\mathcal{L}_{18}$	$\mathcal{L}_{19}$	$\mathcal{L}_1$	$\mathcal{L}_7$
3	$\mathcal{L}_8$	$\boldsymbol{\mathcal{L}_2}$	$\mathcal{L}_5$	$\mathcal{L}_3$

Fig. 9. Latin Cube for Example 1, with B's transmitted symbols along the rows and C's transmitted symbols along the columns

---

Algorithm 1: Obtaining the Latin Cube of Second Order from the constrained $4 \times 4 \times 4$ array

---

Input: The constrained $4 \times 4 \times 4$ array
Output: A Latin Cube of Second Order representing the clustering map at the relay

Start with the constrained $4 \times 4 \times 4$ array
Initialize all empty cells of $\mathcal{X}$ to 0
Let $\mathcal{Y}$ denote the 4×16 matrix obtained by concatenating $\mathcal{X}$ row-wise, and let $\mathcal{Z}$ denote the 16×4 matrix obtained by concatenating $\mathcal{X}$ column-wise
The $(i, j, k)^{th}$ element of $\mathcal{X}$ is the i^{th} file, the j^{th} row and the k^{th} column cell.
for $1 \leq i \leq 4$ **do**
    for $1 \leq j \leq 4$ **do**
        for $1 \leq k \leq 4$ **do**
            if *cell* (i, j, k) *of* $\mathcal{X}$ *is NULL* **then**
                Initialize c=1
                if $\mathcal{L}_c$ *does not occur in the* i^{th} *file of* $\mathcal{X}$*, the* j^{th} *row of* $\mathcal{Y}$ *and the* k^{th} *column of* $\mathcal{Z}$ **then**
                    replace 0 at cell (i, j, k) of $\mathcal{X}$ with $\mathcal{L}_c$;
                    replace $\mathcal{Y}$ with the 4×16 matrix obtained by concatenating $\mathcal{X}$ row-wise, and $\mathcal{Z}$ by the 16×4 matrix obtained by concatenating $\mathcal{X}$ column-wise;
                else
                    c=c+1;
                end
            end
        end
    end
end

$\{(0,0,3),(3,3,0)\}$, $\{(0,1,2),(3,2,1)\}$, $\{(0,1,3),(3,2,0)\}$, $\{(1,0,2),(2,3,1)\}$, $\{(1,0,3),(2,3,0)\}$, $\{(1,1,2),(2,2,1)\}$, $\{(1,1,3),(2,2,0)\}$ must be the same. Similarly the other constraints can be obtained. The constrained array is shown in Figure 10, and the clustering is as shown in Figure 11.

For each of the 112 possibilities of singular fade subspaces of *Case 3* of the previous section, a clustering of size between 16 to 23 can be achieved by first constraining the array representing the relay map in order to remove the singular fade state and then completing the constrained array using the provided algorithm.

V. SIMULATION RESULTS

The proposed scheme is based on the removal of singular fade states for the three-way relaying sccenario. A minimum cluster distance greater than zero is ensured for all the fade states, excluding a subset of singular fade states, which are shown to be non-removable. It is attempted to ensure that in the given scenario, the number of clusters in the clustering, which is the same as the size of the signal set used during the BC phase is minimized. Simulation results presented in this section identify some cases where the proposed scheme outperforms the naive approach that uses the same map for all fade states and vice verse All the simulation results shown in this section are for the case when the end nodes use 4-PSK signal set. The simulation results for the end to end BER as a function of SNR are presented in this section for different fading scenarios.

Consider the case when $H_A, H_B, H_C, H'_A, H'_B$ and H'_C are distributed according to Rician distribution and channel variances equal to 0 dB. The SNR vs BER curve for this case, for a frame length of 256 bits. The plots for the cases with a Rician Factors of 5 dB, 10 dB, 15 dB and 20 dB are as shown in Fig. 13, Fig. 14, Fig. 15 and Fig. 16 respectively. The figures show the SNR vs bit-error-rate curves for the following schemes: adaptive clustering given in the paper, and non-adaptive clustering. For non-adaptive clustering, the relay node uses the same map given by Figure 12 for all the channel realisations, we refer to this map as the non-adaptive map. It can be seen from Fig. 14 that the schemes based on the adaptive clustering relaying perform better than the schemes based on non-adaptive clustering at low SNR, since adaptive clustering removes 112 singular fade states.

It can be seen from the simulation results presented that the non-adaptive network coding performs better than the proposed scheme above a certain SNR when there is a dominant line of sight component, as in the case of Rician

$x_A = 0$	0	1	2	3
0		$\mathcal{L}_5$	$\mathcal{L}_1$	$\mathcal{L}_2$
1			$\mathcal{L}_3$	$\mathcal{L}_4$
2				
3		$\mathcal{L}_6$	$\mathcal{L}_7$	

$x_A = 1$	0	1	2	3
0			$\mathcal{L}_4$	$\mathcal{L}_3$
1	$\mathcal{L}_5$		$\mathcal{L}_2$	$\mathcal{L}_1$
2	$\mathcal{L}_6$			$\mathcal{L}_7$
3				

$x_A = 2$	0	1	2	3
0				
1	$\mathcal{L}_7$			$\mathcal{L}_6$
2	$\mathcal{L}_1$	$\mathcal{L}_2$		$\mathcal{L}_5$
3	$\mathcal{L}_3$	$\mathcal{L}_4$		

$x_A = 3$	0	1	2	3
0		$\mathcal{L}_7$	$\mathcal{L}_6$	
1				
2	$\mathcal{L}_4$	$\mathcal{L}_3$		
3	$\mathcal{L}_2$	$\mathcal{L}_1$	$\mathcal{L}_5$	

Fig. 10. Constraints for Example 1, with B's transmitted symbols along the rows and C's transmitted symbols along the columns

$x_A = 0$	0	1	2	3
0	$\mathcal{L}_8$	$\boldsymbol{\mathcal{L}_5}$	$\boldsymbol{\mathcal{L}_1}$	$\boldsymbol{\mathcal{L}_2}$
1	$\mathcal{L}_9$	$\mathcal{L}_{10}$	$\boldsymbol{\mathcal{L}_3}$	$\boldsymbol{\mathcal{L}_4}$
2	$\mathcal{L}_{11}$	$\mathcal{L}_{12}$	$\mathcal{L}_{13}$	$\mathcal{L}_{14}$
3	$\mathcal{L}_{15}$	$\boldsymbol{\mathcal{L}_6}$	$\boldsymbol{\mathcal{L}_7}$	$\mathcal{L}_{16}$

$x_A = 1$	0	1	2	3
0	$\mathcal{L}_{10}$	$\mathcal{L}_9$	$\boldsymbol{\mathcal{L}_4}$	$\boldsymbol{\mathcal{L}_3}$
1	$\boldsymbol{\mathcal{L}_5}$	$\mathcal{L}_8$	$\boldsymbol{\mathcal{L}_2}$	$\boldsymbol{\mathcal{L}_1}$
2	$\boldsymbol{\mathcal{L}_6}$	$\mathcal{L}_{15}$	$\mathcal{L}_{16}$	$\boldsymbol{\mathcal{L}_7}$
3	$\mathcal{L}_{12}$	$\mathcal{L}_{11}$	$\mathcal{L}_{14}$	$\mathcal{L}_{13}$

$x_A = 2$	0	1	2	3
0	$\mathcal{L}_{13}$	$\mathcal{L}_{14}$	$\mathcal{L}_{11}$	$\mathcal{L}_{12}$
1	$\boldsymbol{\mathcal{L}_7}$	$\mathcal{L}_{16}$	$\mathcal{L}_{15}$	$\boldsymbol{\mathcal{L}_6}$
2	$\boldsymbol{\mathcal{L}_1}$	$\boldsymbol{\mathcal{L}_2}$	$\mathcal{L}_8$	$\boldsymbol{\mathcal{L}_5}$
3	$\boldsymbol{\mathcal{L}_3}$	$\boldsymbol{\mathcal{L}_4}$	$\mathcal{L}_9$	$\mathcal{L}_{10}$

$x_A = 3$	0	1	2	3
0	$\mathcal{L}_{16}$	$\boldsymbol{\mathcal{L}_7}$	$\boldsymbol{\mathcal{L}_6}$	$\mathcal{L}_{15}$
1	$\mathcal{L}_{14}$	$\mathcal{L}_{13}$	$\mathcal{L}_{12}$	$\mathcal{L}_{11}$
2	$\boldsymbol{\mathcal{L}_4}$	$\boldsymbol{\mathcal{L}_3}$	$\mathcal{L}_{10}$	$\mathcal{L}_9$
3	$\boldsymbol{\mathcal{L}_2}$	$\boldsymbol{\mathcal{L}_1}$	$\boldsymbol{\mathcal{L}_5}$	$\mathcal{L}_8$

Fig. 11. Clustering for Example 2

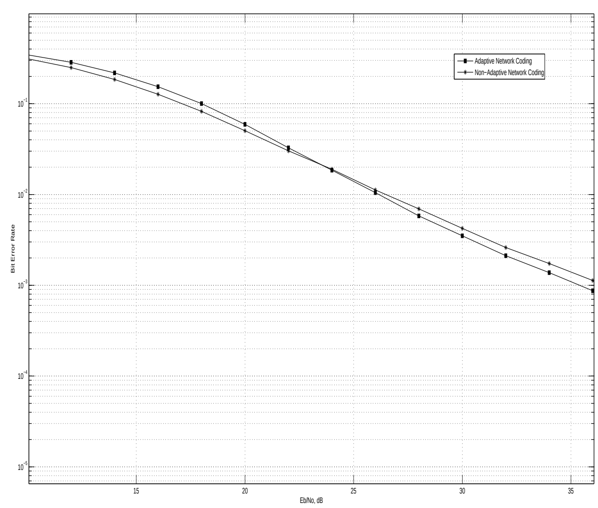

Fig. 13. SNR vs ber curves for different schemes for 4-PSK signal set when the Rician Factors is 5 dB

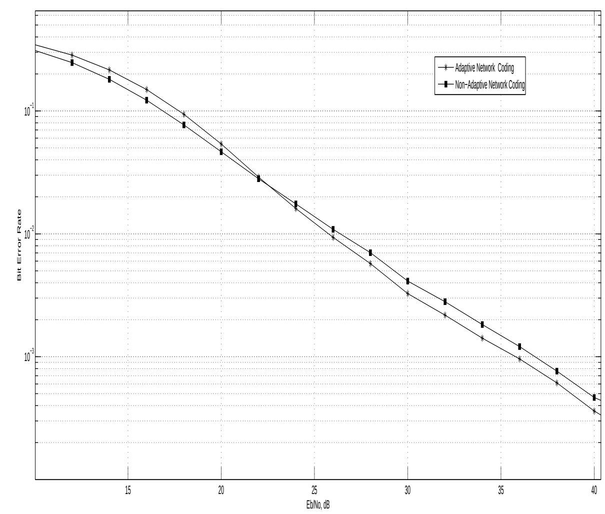

Fig. 14. SNR vs ber curves for different schemes for 4-PSK signal set when the Rician Factors is 10 dB

fading scenario. The reason for this is as follows: The end to end SNR vs BER as well as the throughput performance depend on the performance during the MA phase as well as the BC phase. As the line of sight component becomes more and more predominant, the performance during the BC phase gets better and better, but the effect of multiple access interference which occurs during the MA phase remains the same. Hence, for the cases when line of sight component is predominant, the performance degradation due to the MA interference predominates over the degradation occurring during the BC phase. The case of non-adaptive network coding, the relay utilizes a constellation of least possible size, whereas in adaptive network coding, the relay attempts at removing singular fade states, thereby optimizing the performance during MA phase to the fullest extent possible, at the cost of degraded performance during the BC phase. Hence, the non-adaptive network coding performs better than the proposed scheme at high SNR.

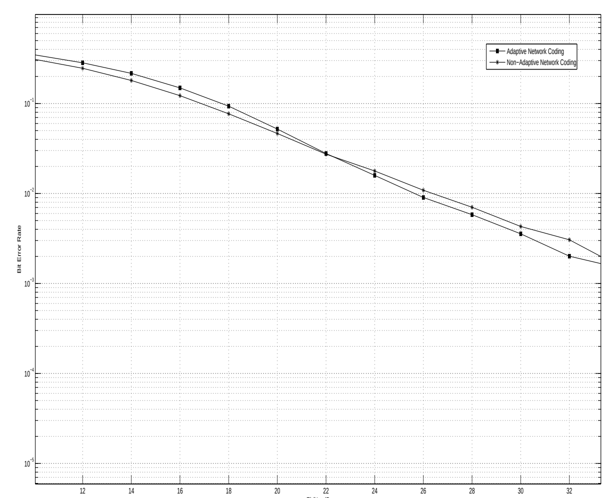

Fig. 15. SNR vs ber curves for different schemes for 4-PSK signal set when the Rician Factors is 15 dB

$x_A = 0$	0	1	2	3
0	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	$\mathcal{L}_4$
1	$\mathcal{L}_5$	$\mathcal{L}_6$	$\mathcal{L}_7$	$\mathcal{L}_8$
2	$\mathcal{L}_9$	$\mathcal{L}_{10}$	$\mathcal{L}_{11}$	$\mathcal{L}_{12}$
3	$\mathcal{L}_{13}$	$\mathcal{L}_{14}$	$\mathcal{L}_{15}$	$\mathcal{L}_{16}$

$x_A = 1$	0	1	2	3
0	$\mathcal{L}_6$	$\mathcal{L}_5$	$\mathcal{L}_8$	$\mathcal{L}_7$
1	$\mathcal{L}_2$	$\mathcal{L}_1$	$\mathcal{L}_4$	$\mathcal{L}_3$
2	$\mathcal{L}_{14}$	$\mathcal{L}_{13}$	$\mathcal{L}_{16}$	$\mathcal{L}_{15}$
3	$\mathcal{L}_{10}$	$\mathcal{L}_9$	$\mathcal{L}_{12}$	$\mathcal{L}_{11}$

$x_A = 2$	0	1	2	3
0	$\mathcal{L}_{11}$	$\mathcal{L}_{12}$	$\mathcal{L}_9$	$\mathcal{L}_{10}$
1	$\mathcal{L}_{15}$	$\mathcal{L}_{16}$	$\mathcal{L}_{13}$	$\mathcal{L}_{14}$
2	$\mathcal{L}_3$	$\mathcal{L}_4$	$\mathcal{L}_1$	$\mathcal{L}_2$
3	$\mathcal{L}_7$	$\mathcal{L}_8$	$\mathcal{L}_5$	$\mathcal{L}_6$

$x_A = 3$	0	1	2	3
0	$\mathcal{L}_{16}$	$\mathcal{L}_{15}$	$\mathcal{L}_{14}$	$\mathcal{L}_{13}$
1	$\mathcal{L}_{12}$	$\mathcal{L}_{11}$	$\mathcal{L}_{10}$	$\mathcal{L}_9$
2	$\mathcal{L}_8$	$\mathcal{L}_7$	$\mathcal{L}_6$	$\mathcal{L}_5$
3	$\mathcal{L}_4$	$\mathcal{L}_3$	$\mathcal{L}_2$	$\mathcal{L}_1$

Fig. 12. Non-Adaptive map

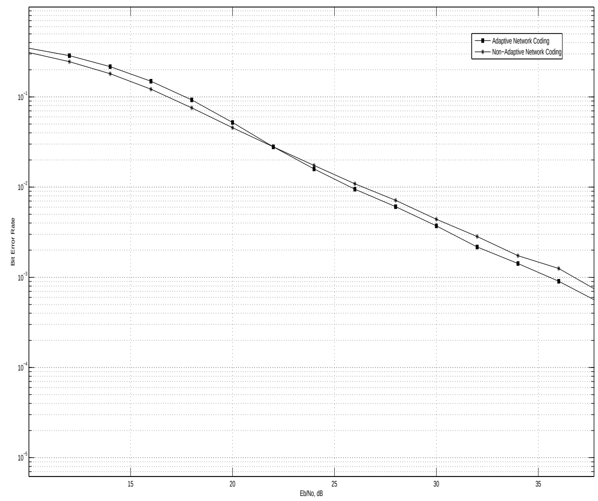

Fig. 16. SNR vs ber curves for different schemes for 4-PSK signal set when the Rician Factors is 20 dB

VI. Conclusion

Our paper deals with the three-way wireless relaying scenario, assuming that the three nodes operate in half-duplex mode and that they transmit points from the same 4-PSK constellation. It is shown that it is possible for information exchange to take place using just two channels uses, unlike the other work done for the case, to the best of our knowledge. The Relay node clusters the 4^3 possible transmitted tuples (x_A, x_B, x_C) into various clusters such that *the exclusive law* is satisfied. This necessary requirement of satisfying the exclusive law is shown to be the same as the clustering being represented by a Latin Cube of second order. Using the proposed schemes, not only is the exchange of information between the three nodes made possible using three channel uses, the size of the resulting constellation used by the relay node R in the BC phase is reduced from 4^3 to lie between 16 to 23. Note that we do not claim that the size of the clustering utilizing modified clustering is the best that can be achieved, since our method of filling the Latin Cube of Second Order of side 4 may not be the most optimal process of doing so, and it might be possible to fill the array with less than 23 symbols.

Acknowledgement

This work was supported partly by the DRDO-IISc program on Advanced Research in Mathematical Engineering through a research grant as well as the INAE Chair Professorship grant to B. S. Rajan.

References

[1] S. Zhang, S. C. Liew and P. P. Lam, “Hot topic: Physical-layer Network Coding”, ACM MobiCom ’06, pp. 358–365, Sept. 2006.

[2] S. J. Kim, P. Mitran and V. Tarokh, “Performance Bounds for Bidirectional Coded Cooperation Protocols”, IEEE Trans. Inf. Theory, Vol. 54, pp. 5235–5241, Nov. 2008.

[3] P. Popovski and H. Yomo, “Physical Network Coding in Two-Way Wireless Relay Channels”, IEEE ICC, Glasgow, Scotland, pp. 707–712, June 2007.

[4] T. Koike-Akino, P. Popovski and V. Tarokh, “Optimized constellation for two-way wireless relaying with physical network coding”, IEEE Journal on selected Areas in Comm., Vol.27, pp. 773–787, June 2009.

[5] T. Koike-Akino, P. Popovski and V. Tarokh, “Denoising strategy for convolutionally-coded bidirectional relaying”, IEEE ICC 2009, Dresden, Germany, June 2009.

[6] B. Hern and K. Narayanan, “Multilevel Coding Schemes for Compute-and-Forward”, IEEE ISIT, St. Petersburg, Russia, pp. 1-5, July 2011.

[7] Vishnu Namboodiri, Vijayvaradharaj T Muralidharan and B. Sundar Rajan, “Wireless Bidirectional Relaying and Latin Squares”, available online at arXiv:1110.0084v2 [cs.IT], 1 Oct. 2011.

[8] Chris A. Rodger, “Recent Results on The Embedding of Latin Squares and Related Structures, Cycle Systems and Graph Designs”, Le Matematiche, Vol. XLVII - Fasc. II, pp. 295–311, 1992.

[9] C.H. Lui and A. Arapostathis, “Joint network coding and superposition coding for multi-user information exchange in wireless relaying networks”, in Proc. of IEEE-Globecom, pp. 1–6, Dec. 2008.

[10] Mylene Pischella and Didier Le Ruyet, “Lattice based coding scheme for MIMO bi-directional relaying with three nodes”, 22nd IEEE Personal Indoor Mobile Radio Communications, Toronto, Canada, pp. 1459–1463, Sept. 2011.

[11] Moonseo Park and Seong Keun Oh, “An Iterative Network Code Optimization for Three-Way Relay Channels”, Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th, pp. 1–5, Sept. 2009.

[12] Youngil Jeon, Young-Tae Kim, Moonseo Park, Inkyu Lee, “Opportunistic Scheduling for Three-way Relay Systems with Physical Layer Network Coding”, Vehicular Technology Conference (VTC Spring) 2011, Budapest, Hungary, IEEE 73rd, pp. 1–5, May 2011.

[13] K. Kishen, “On Latin and Hyper-Graeco-Latin Cubes and Hyper Cubes”, Current Science, Vol. 11, pp. 98–99, 1942.

[14] Vijayvaradharaj T Muralidharan, Vishnu Namboodiri and B. Sundar Rajan, “Channel Quantization for Physical-Layer Network Coded Two-Way Relaying”, available online at arXiv:1109.6101v1 [cs.IT], 28 Sept. 2011.